

Cite No. 2

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 4 : G06F 13/38, 13/42, 15/56		A1	(11) International Publication Number: WO 89/09967
			(43) International Publication Date: 19 October 1989 (19.10.89)
(21) International Application Number: PCT/US89/01456			(81) Designated States: AT (European patent), AU, BE (European patent), CH (European patent), DE (European patent), FR (European patent), GB (European patent), IT (European patent), KR, LU (European patent), NL (European patent), SE (European patent). Published With international search report.
(22) International Filing Date: 7 April 1989 (07.04.89)			
(30) Priority data: 179,412 8 April 1988 (08.04.88) US			
(71) Applicant: COGENT RESEARCH, INC. (US/US); 1100 N.W. Compton Drive, Beaverton, OR 97006 (US).			
(72) Inventors: VOLLUM, Charles, A.; 24850 S.W. Ladd Hill Road, Sherwood, OR 97140 (US); HENSON, Noel, W.; 2884 S.E. 43rd Court, Hillsboro, OR 97123 (US).			
(74) Agent: SMITH-HILL, John; Delfert, Smith-Hill and Bedell, 1070 One Main Place, 101 S.W. Main, Portland, OR 97204 (US).			
(54) Title: COMPUTING MACHINE WITH HYBRID COMMUNICATION ARCHITECTURE			
(57) Abstract			
<p>A computer comprises a plurality of processor modules (110₁-110_N), each having at least first and second I/O connection interfaces (126, 130A, 130B, 130C, 130D), a processor (122) connected to those interfaces (126, 130A, 130B, 130C, 130D), and read-write memory (118) connected to the processor. The first I/O connection interface (126) of each processor module (110₁-110_N) is connected to a common bus (138). The second I/O connection interface (130A, 130B, 130C, 130D) of each processor module (110₁-110_N) is connected to a switch (150A, 150B, 150C, 150D), the switch (150A, 150B, 150C, 150D) being operative to connect the second I/O connection interface (130A, 130B, 130C, 130D) of a selected processor module (110₁-110_N) selectively to the second I/O connection interface (130A, 130B, 130C, 130D) of any other processor module (110₁-110_N). A controller (142) is connected to the bus (138) for receiving over the bus (138) data pertaining to a first processor module (110₁-110_N), which requires access to information, and to a second processor module (110₁-110_N), from which the required information is available, and for controlling the switch (150A, 150B, 150C, 150D) to allow the required information to be transmitted from the first processor module (110₁-110_N) to the second processor module (110₁-110_N).</p>			

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FR	France	ML	Mali
AU	Australia	GA	Gabon	MR	Mauritania
BB	Barbados	GB	United Kingdom	MW	Malawi
BE	Belgium	HU	Hungary	NL	Netherlands
BG	Bulgaria	IT	Italy	NO	Norway
BJ	Benin	JP	Japan	RO	Romania
BR	Brazil	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic	KR	Republic of Korea	SE	Sweden
CG	Congo	LI	Liechtenstein	SN	Senegal
CH	Switzerland	LU	Luxembourg	SU	Soviet Union
CM	Cameroun	LV	Latvia	TD	Chad
DE	Germany, Federal Republic of	MC	Monaco	TG	Togo
DK	Denmark	MG	Madagascar	US	United States of America
FI	Finland				

WO 89/09967

PCT/US89/01456

1

COMPUTING MACHINE WITH
HYBRID COMMUNICATION ARCHITECTURE

Background of the Invention

5 This invention relates to a computing machine
with hybrid communication architecture.

A digital computer solves a problem by break-
ing the problem down into multiple steps. A compu-
ter with a single processor is able to execute one
step at a time. It takes an inordinate time to
10 solve a complex problem by use of such a sequential
mode of operation. By operating multiple proces-
sors in parallel, it is generally possible to
reduce substantially the time required to solve a
problem.

15 If multiple processors are operated in
parallel, it is necessary for the processors to
share data. One technique for sharing data among
multiple processors is for there to be a common, or
global, memory to which all the processors have
20 access on an equal footing. A problem with this
technique arises from the fact that only one pro-
cessor can access the memory at a given time, and
therefore contention problems limit the number of
processors that can be accommodated. The number of
25 processors can be increased somewhat by use of
coherent caching or crossbar switching, but such
techniques are costly and cumbersome.

A second method of allowing sharing of data by
multiple processors involves use of a parallel
30 communication bus. A bus allows a great deal of
flexibility in communication, including the ability
to broadcast data from one processor to many or all
of the others in a single operation. Each proces-
sor is able to execute independently until the need

35

WO 89/09967

PCT/US89/01456

2

to communicate arises. However, when communication is necessary, contention problems arise since only one processor can transmit on the bus at a time. Use of multiple buses can reduce problems due to contention, but multiple buses also reduce flexibility and add greatly to cost and complexity.

A third technique for allowing multiple processors to share data is provided by point-to-point communication links. Processors that have built-in links are commercially available, and therefore they are very easy to provide. Links offer virtually unlimited expansion possibilities, since the number of communication paths increases whenever a processor is added. However, links are the most difficult to use, since the physical interconnection pattern must match the pattern of communication required by the program that is being executed. If processors are added to or removed from the system, a new pattern of link connections must be established, and the program must be rewritten, recompiled or, at the very least, relinked to match. Broadcasting a message is difficult and time consuming, since the message must be copied from one processor to the next until it has reached all processors. Since two different programs will generally require two different physical interconnection patterns, it has not hitherto been possible to execute multiple programs simultaneously using links for communication between processors unless the programs are specifically designed to require the same communication patterns.

The difficulty of matching the physical interconnection pattern with the pattern of communication required by the program is partially

WO 89/09967

PCT/US89/01456

4

5 The read operation is identical to in except that the matching tuple is not removed from the tuple space. The eval operation is a specialized form of out. Out creates a passive tuple, whereas eval creates an active tuple.

10 If a processor performs an in operation, it is necessary to search tuple space for the matching tuple. It would be time consuming to examine each tuple in turn to determine whether it matches the template, and therefore the tuples are classified and a directory is created to facilitate the search. In a multi-processor computer, different portions of the directory are accessed by the different processors, and in order to complete an
15 in operation, potentially all the processors must examine their portions of the directory against the template in order to determine whether a matching tuple exists.

20 The Linda system makes the programmer's job vastly easier; since he need not know the source or the destination of his data. The directory is automatically consulted and used to match a request for a tuple with a tuple that is available. When a required tuple is not available, the requester
25 waits until such a tuple becomes available. The Linda system can be implemented on any of the above-mentioned parallel processing architectures, most efficiently with global memory, next most efficiently with one or more buses, and least
30 efficiently with links.

35 Maintenance of the Linda distributed directory is well suited to the bus, because it enables notification to be broadcast to potentially interested processors as tuples becomes available. The notification messages are short, and therefore a

WO 89/09967

PCT/US89/01456

3

alleviated by adding a link crossbar switch, by which the configuration of the physical interconnection pattern can easily be changed at the start of a program to match the communication pattern required by the program. However, use of a link crossbar switch in this manner does not ease the job of defining a suitable connection pattern, nor does it facilitate simultaneous execution of two or more programs using the same set of processors.

10 D. Gelernter, Generative Communication in Linda, ACM Trans. Prog. Lang. and Sys., Vol. 7, No. 1, pages 80-112, (1985) describes a software system called Linda. Linda is based on the use of tuples. A tuple is a collection of related data. Elements of a tuple are fields holding actual values or formalis. There are two types of tuples, namely passive tuples and active tuples. A passive tuple is simply a collection of data items, whereas an active tuple is a process which becomes a passive tuple. Tuples exist in an abstract space called tuple space. The tuple space may exist over multiple processors. Four principal operations can be performed on tuple space. The out operation is an operation that creates a tuple and places it in tuple space. The in operation is the reverse of out: it specifies a tuple that it desires, in the form of a template, and the computer matches the template against all the tuples existing in tuple space. If a matching tuple is found, it is removed from tuple space and is returned to the requesting process. When no tuple matches, the in operation blocks, and the requesting process is suspended until another process, through an out operation, creates a matching tuple. At this point, the requesting process continues. An out operation can never block.

WO 89/09967

PCT/US89/01456

5

great volume of them can be moved over the bus in a short period. However, if a tuple, which might contain a large block of data, is moved over the bus, updates to the directory may be blocked for long periods, resulting in poor system performance.

Using links to maintain the Linda directory is very inefficient, since broadcasting requires repeated passing of the message from processor to processor. However, moving data blocks from producer to requester is an operation to which links are well suited. Since there are many links, many such transfers can be taking place simultaneously if the right link connections are made.

Other publications that refer to Linda include D. Gelernter, Programming for Advanced Computing, Scientific American, October, 1987, pages 91-98; R. Bjornson, N. Carriero, D. Gelernter and J. Leichter, Linda, the Portable Parallel, (1988); N. Carriero and D. Gelernter, Applications Experience with Linda, (1988); N. Carriero, D. Gelernter and J. Leichter, Distributed Data Structures in Linda, Proc. ACM Symp. on Principles of Prog. Lang., pages 236-242 (1986); D. Gelernter, N. Carriero, S. Chandren and S. Chang, Parallel Programming in Linda, Proc. Int. Conf. on Parallel Processing, pages 255-263 (1985); N. Carriero and D. Gelernter, The S/Net's Linda Kernel, ACM Trans. Comp. Sys., Vol. 4, No. 2, pages 110-129 (1986); N. Carriero, Implementation of Tuple Space Machines, Yale University, Dept. of Computer Science, Research Report YALEU/DCS/RR-567 (1987); N. Carriero and D. Gelernter, Integrating Multiple Tuple Spaces, the File System and Process Management in a Linda-based Operating System, Yale University, Dept. of Computer Science, Research

WO 89/09967

PCT/US89/01456

6

Report YALEU/DCS/RR-Technical Memo (1988); M. Factor and D. Galernter, The Parallel Process Lattice as an Organizing System for Realtime Knowledge Daemons, _____ (1988).

5

Summary of the Invention

A preferred embodiment of the present invention is a computer which comprises a plurality of processor modules, each having at least first and second I/O connection interfaces, a processor connected to those interfaces, and memory connected to the processor. The first I/O connection interface of each processor module is connected to a common bus. The second I/O connection interface of each processor module is connected to a switch, the switch being operative to connect the second I/O connection interface of a selected processor module selectively to the second I/O connection interface of any other processor module. A controller is connected to the bus for receiving over the bus data pertaining to a first processor module, which requires access to information, and to a second processor module, from which the required information is available, and for controlling the switch to allow the required information to be transmitted from the first processor module to the second processor module by way of the switch.

In a computer embodying the present invention, the Linda directory can be updated by broadcasting a template over the bus, and when a processor module that has access to a matching tuple is identified, the tuple is transmitted from the producing processor to the requesting processor in a non-broadcast fashion, by way of the switch. During the transmission, the bus is available to process

WO 89/09967

PCT/US89/01456

7

other updates and arrange other connections, so that many transmissions can occur simultaneously.

Brief Description of Drawings

5 For a better understanding of the invention, and to show how the same may be carried into effect, reference will now be made, by way of example, to the accompanying drawings in which;

10 FIG. 1 is a block diagram of a first networked computer system embodying the present invention,

FIG. 2 is a more detailed block diagram of one of the computer stations shown in FIG. 1,

15 FIG. 3 is a simplified block diagram of a second networked computer system embodying the invention, and

FIG. 4 is a simplified block diagram of a stand-alone computer embodying the invention.

Detailed Description

20 The computer system illustrated in FIG. 1 comprises a user terminal 10, several computer stations 12 and a disc drive station 14. The user terminal 10 comprises a processor 16 which is connected to various user utilities, such as a
25 display card 18, a hard and/or floppy disc drive card 20, and a keyboard card 24, through its memory bus 26. The memory bus is also connected to local random access memory (RAM) 28 and local read only memory (ROM) 30. The processor 16 has four link
30 interfaces 44. A second processor 42 is connected over its memory bus to local RAM 48 and has four link interfaces 52. In the preferred embodiment of the present invention, each processor 16, 42 is an Inmos IMS T800 transputer. One link interface of
35 the processor 16 is connected to a link interface

WO 89/09967

PCI/US89/01456

8

of the processor 42, and the other three link interfaces of each processor are connected to respective demultiplexed interfaces of a 6:1 byte domain multiplexer/demultiplexer (mux/demux) 60, which has a multiplexed interface connected to a fiber optic transceiver 64. The mux/demux 60 and the fiber optic transceiver 64 are used to transmit data and instructions between the processors 16, 42 and a fiber optic cable 68, which connects the user terminal 10 to one of the computer stations 12. Messages are transmitted over the cable 68 at a rate of about 100 Mb/s.

The mux/demux 60 has a multiplexer channel and a demultiplexer channel. The multiplexer channel of the mux/demux comprises, for each demultiplexed interface, a serial to parallel converter which receives serial data over its link in words of eight bits at a rate 10-20 Mb/s and, for each eight-bit serial word, generates an eight-bit parallel word and applies it to a parallel bus with a four-bit tag that designates the particular demultiplexed interface that provided the serial word. Thus, for each eight-bit word received at a demultiplexed interface, a twelve-bit word is applied to the parallel bus. The parallel bus is connected to a high speed parallel-to-serial converter, which reads the twelve-bit parallel words in turn and generates an electrical signal composed of a succession of twelve-bit serial words. The electrical signal provided by the parallel-to-serial converter is applied to the fiber optic transceiver 64.

The demultiplexer channel of the mux/demux 60 comprises a serial-to-parallel converter which receives twelve-bit serial words from the fiber

WO 89/09967

PCT/US89/01456

9

5 optic transceiver 64 and generates twelve-bit parallel words which are applied to the parallel bus. Each twelve-bit word comprises a four-bit tag and an eight-bit data word. Six parallel-to-serial
10 converters, which are connected to the demultiplexed interfaces respectively, read the parallel bus. The parallel-to-serial converter that is to read a particular data word is designated on the basis of the four-bit tag associated with the data
15 word. Thus, in the transmit mode the mux/demux receives up to six serial signals and interleaves them to provide a single serial output signal at its multiplexed interface. The output signal of the mux/demux is applied to the fiber optic transceiver 64, which launches an optical signal, coded in accordance with the signal provided by the
20 mux/demux, into the fiber optic cable 68. In the receive mode, a coded optical signal is received over the fiber optic cable and the fiber optic transceiver generates a serial electrical signal in response thereto. The serial electrical signal is applied to the multiplexed interface of the
25 mux/demux 60 and the mux/demux demultiplexes it into up to six signals which are provided at the demultiplexed interfaces respectively of the mux/demux.

30 When data or commands are being input into the system, the processors 16, 42 manipulate the data or commands in accordance with programs stored in the ROM 30 and apply the data or commands in serial fashion over the links, the mux/demux 60 and the transceiver 64 to the fiber optic cable 68. When data or commands are received by the transceiver over the fiber optic cable 68, the serial signal is
35 demultiplexed into up to six serial signals which

WO 89/09967

PCT/US89/01456

10

are applied to the processors 16, 42 over the links, and the processors apply the data or commands to the display card 12 or the disc drive card 20.

5 As shown in FIG. 2, each computer station 12 comprises several processor modules 110, each of which is composed of a processor 114 and a random access memory 118. In the preferred embodiment of the present invention, the processor 114 is an Inmos
10 IMS T800 transputer, comprising a CPU 122, an external parallel interface 126 and four link interfaces 130A-130D. The processor 114 also includes other components, but these are not relevant to an understanding of the present invention and there-
15 fore are not illustrated in the drawings.

The CPU 122, memory 118, parallel interface 126 and link interfaces 130A-130D of each processor 114 communicate with each other over an internal 32-bit parallel bus 134. The parallel interface
20 126 is connected to an external 32-bit parallel bus 138, which is connected to the external parallel interface of each other processor module, a crossbar switch operator 142 and at least one external communication module 144. In the preferred embodi-
25 ment of the present invention, the crossbar switch operator 142 is an Inmos IMS T414 transputer, which has essentially the same architecture as the IMS T800 transputer.

The link interfaces of the processor modules
30 are connected to a programmable crossbar switch 146, which is implemented by four Inmos IMS C004 programmable link switches 150A-150D. Each link switch has 32 data link connections. Respective data link connections of the link switch 150A, for
35 example, are connected to the link interfaces 130A

WO 89/09967

PCT/US89/01456

11

of the processors 114. Each link switch also has a configuration link connection 158 over which it receives a signal for establishing the manner in which the data link connections are connected by the switch. The configuration link connections 158 of the four link switches are connected to respective link interfaces of the operator 142.

The crossbar switch 146, the switch operator 142 and the parallel bus 138 are all carried by a mother board having sixteen connection slots. Each connection slot can receive either a communications card, which carries a communication module 144, or a processor card, which carries two processor modules 110. Therefore, the maximum number of processor modules that can be accommodated (it being necessary to have at least one communication module) is thirty. The four programmable link switches provide an aggregate of 128 link connections, and the maximum of thirty processor modules occupy 120 of these data link connections. The other eight link connections are connected to the external communication module, for purposes which will be described below. If there is more than one communication module (and consequently fewer than thirty processor modules), each communication module is connected to two link connections of each link switch.

Referring again to FIG. 1, the disc drive station 14 comprises a single processor 160 which is connected over its memory bus 168 to a high speed disc drive card 162 and to local RAM and local ROM. The processor 160, which may be an Inmos IMS T800 transputer, has four link interfaces which are connected to a 4:1 mux/demux 164. The mux/demux 164 of the disc drive station 14 is essentially the same as the mux/demux 60 of the

WO 89/09967

PCT/US89/01456

12

user terminal 10, and is connected to a fiber optic transceiver 166. The disc drive station provides high speed disc access without being burdened by the need to generate a display.

5 The computer stations 12 shown in FIG. 1 each have two communication modules 144, which enable the stations 12 to be connected in a linear arrangement between the user terminal and the disc drive station. Each computer station therefore can
10 accommodate up to twenty-eight processor modules 110. Alternatively, the user terminal 10, the disc drive station 14 and the computer stations 12 may be connected in a star arrangement, as shown in FIG. 3. In this arrangement, the computer stations
15 12A-12D each need only one communication module, but the computer station 12E has six communication modules. A third possibility is for the computer stations to be organized as a hierarchical tree. Numerous other arrangements, employing different
20 interconnection schemes among the user terminal, the disc drive station and the necessary processor modules, are possible.

Each computer station 12 executes an application by use of the Linda language. Data are
25 received by the computer station by way of an external communication module 144 and are stored in the external memories 118. The data stored in the memories 118 are associated as tuples, and each external memory 118 includes a directory portion
30 containing information regarding the tuples that are stored in that memory. When the processor 114 of a processor module 110 executes an out operation, the tuple generated in the out operation is loaded into the processor module's external memory
35 and the processor module's portion of the directory

WO 89/09967

PCT/US89/01456

13

is updated to reflect addition of this tuple. When a processor module performs an in operation, it first examines its own portion of the directory against the template that defines the desired tuple. If no match is found, the requesting processor module may broadcast the template over the parallel bus 138 to other processor modules of the computer station. The receiving processor modules examine their respective portions of the directory, and the first processor module that finds a match places a signal on the bus to indicate that the other processor modules should cease the search. The requesting and producing processor modules then provide signals to the operator 142, and the operator responds by causing the switch 146 to establish a connection between a link interface of the requesting processor module and the corresponding link interface of the producing processor module. The matching tuple is then transmitted from the producing processor module to the requesting processor module through the links and the crossbar switch 114, and does not occupy the bus 138.

When an in or read operation takes place, the directory portion of the requesting processor module is updated to reflect the fact that the tuple space of that processor module now contains the specified tuple. Similarly, when an in operation (but not a read operation) is performed, the directory portion of the producing module is updated to reflect the fact that it no longer has the specified tuple in its tuple space. When an out or eval operation is performed, the directory portion of the module that executes the operation is updated. It will therefore be seen that it

WO 89/09967

PCT/US89/01456

14

is not necessary to burden the bus with messages pertaining to the contents of the tuple space of each processor module.

5 Bus operation has four distinct cycles. A processor that requires access to the bus 138 in order to transmit a message asserts a bus request signal on a bus control line and an arbitration cycle takes place. If no other processor requires access to the bus at that time, the first-mentioned
10 processor wins the arbitration by default. If one or more other processors requires access to the bus, distributed arbitration logic ensures fair arbitration among the processors that require access. When the arbitration cycle is complete, a
15 selection cycle takes place. The transmitting processor writes a single 32-bit word onto the bus. If the computer station has twenty-eight processor modules and two communication modules, twenty-eight bits of this word define, on a one bit per module
20 basis, a mask of the processor modules that are to receive the ensuing message. Two more bits determine whether the external communication modules are to receive the message. This is the selection operation. Thus, with a single bus cycle, the
25 transmitting processor can select any one or more of the other processors to receive its message.

Each processor that is selected by the transmitting processor to receive its message receives an interrupt from its parallel interface. The interrupt
30 forces the receiving processor into a receive message mode, in which each receiving processor reads the parallel bus. The transmitting processor receives a status bit that indicates whether a receiving processor is in the receive message mode, and does not transmit
35 data until all the receiving processors are in the

WO 89/09967

PCT/US89/01456

15

receive message mode. When all the selected processors are in the receive message mode, the transmitting processor and the receiving processors are interlocked and the selection cycle is complete. A transmission cycle then occurs, in which the transmitting processor transmits its message over the bus in words of 32 bits. The transmitting processor holds its data word on the bus until it has received an acknowledgement bit from each receiving processor. The acknowledgement bit indicates that a receiving processor has read the data word from the bus. The transmitting processor then ends its write cycle and each receiving processor ends its read cycle. The transmitting processor then enters another write cycle, in which the next data word is transmitted. The first word placed on the bus during the transmission phase represents the number of words to be transmitted. The receiving processors count the number of words actually transmitted, and when the number transmitted is equal to the number represented by the first word, they process the message. The transmitting processor enters a disconnect cycle in which it negates its bus request, and this allows arbitration to take place again. The processors that were previously selected are deselected. The transmission cycle is then complete, and the bus is available for another transmission.

In the system described, transmissions are accomplished without use of FIFO buffers. However, this is done at the expense of requiring the receiving processors to all operate synchronously during a transmission, and this might not be desirable. Therefore, each processor module could include a FIFO buffer for receiving and temporarily

WO 89/09967

PCT/US89/01436

16

holding data transmitted to that processor module over the bus. In this fashion, the receiving processors are able to run independently of each other. Of course, a FIFO buffer has limited capacity, and if one or more of the buffers is filled during a transmission, it would be necessary to fall back on the previously-described mode of transmitting data from a transmitting processor to the receiving processors.

Each link is composed of two lines, allowing bi-directional communication between two processors. The producing processor transmits words of eleven bits serially over the link using one line, the first bit of each word being a start bit and the last two bits being an ending. On receipt of the first bit of a word, the requesting processor transmits an acknowledgment to the producing processor over the other line of the link. The acknowledgment code is a unique two-bit code and is received by the producing processor before it completes transmitting its word. In fact, the length of the serial word is such that the acknowledgment code can travel through three link switches and still be received by the producing processor, indicating that the next word can be sent, before the transmission of the first word is completed. Accordingly, the producing processor can send a second word immediately after the first word, without having to wait until after the end of the first word to receive an acknowledgement.

By use of the hybrid communication architecture that has been described and illustrated, the computer station shown in FIG. 2 is able to reconfigure the pattern of link connections among its processors dynamically, in response to changes in

WO 89/09967

PCT/US89/01456

17

the communication pattern required by the program that is being executed. Also, multiple programs can be run simultaneously, without regard to the communications patterns required by the programs respectively, since the pattern of link connections is not fixed at the start of a program. The programmer need not consider the communications pattern that will be required by a program that he is writing, since a suitable pattern of link connections is established automatically.

The user terminal 10 can run the same programs as the computer stations 12. However, when it is used as the terminal for a network, the processors 16 and 42 do not run applications but are concerned with graphics, creating the display, reading the keyboard and accessing the disc. The disc drive station 14 is particularly useful if disc access cannot be accomplished by the user terminal 10 with sufficient speed when it is having to perform other functions.

As noted previously, each computer station 12 includes at least one communication module 144. As shown in FIG. 2, each communication module comprises a processor 170, such as an Inmos IMS T800 transputer, having a parallel bus interface and four link interfaces. The parallel bus interface is connected to the parallel bus 138 and the four link interfaces are connected to respective link interfaces of a 12:1 mux/demux 172. The other eight link interfaces of the mux/demux are connected to eight link connections of the switch 146. Except for the number of link interfaces, and consequently the potential maximum speed of operation, the mux/demux 172 of the communication module operates in essentially the same way as the

WO 89/09967

PCT/US89/01456

18

mux/demux 60 of the user terminal 10. The communication modules enable the computer stations to exchange tuple requests and tuples.

5 When a requesting processor of a computer station 12 generates a mask that defines the modules that are to receive the template defining a desired tuple, the mask will generally include the communication module(s) of that station. The message that is transmitted by the requesting processor
10 module includes the name of the tuple space that would contain the desired tuple, if it exists. The communication module stores the names of the tuple spaces that are associated with each of the other computer stations, and if any of the other computer
15 stations with which that communication module can communicate by way of its transceiver 174 is associated with the named tuple space, the communication module will transmit the template over the fiber optic cable. If the message is received by a
20 computer station that includes a processor that is associated with the named tuple space, the processor of the receiving communication module directs the message over the parallel bus 138 to the appropriate processor module(s). If the message is
25 received at one communication module of a processor having a second communication module that is connected to a computer station that includes a processor that is associated with the named tuple space, the processor of the receiving communication module directs the message over the parallel
30 bus 138 to the other communication module for retransmission. Ultimately, the message reaches all computer stations having processors that are associated with the named tuple space.

35 When a matching tuple is found in a computer

WO 89/09967

PCT/US89/01456

19.

station other than the one that contains the requesting processor, the tuple is transmitted through the crossbar switch of the producing computer station to a communication module of that station and is transmitted serially to a communication module of the requesting computer station. This might involve the tuple's passing through one or more intermediate computer stations, and being passed between communication modules of an intermediate station through the crossbar switch of that station. When the tuple reaches the requesting computer station, it is transmitted to the requesting processor module by way of the crossbar switch. Since, as noted previously, a requesting processor issues its acknowledgment code at the beginning of a data word, there is no significant performance penalty with respect to transmission of tuples from computer station to computer station in the described network arrangements.

It will be appreciated that the present invention is not restricted to the particular embodiment that has been described and illustrated, and that variations may be made therein without departing from the scope of the invention as defined in the appended claims and equivalents thereof. For example, the invention is not restricted to use with any particular type of processor. It is necessary only that the processor be able to support two communication modes, one of which is suitable for transmission of data pertaining to information required by the processor and the other of which is suitable for transmission of the information itself. The invention is not restricted to use with computers configured for connection in a network. If the FIG. 1 computer

WO 89/09967

PCT/US89/01456

20

was designed for stand-alone use, it could
accommodate up to thirty-two processor modules.
Further, the invention is not restricted to use of
a crossbar switch to interconnect the link
5 interfaces of the processors. Five processor
modules each having four link interfaces could each
have one link interface hard wired to a link
interface of each other processor module, as shown
in FIG. 4. In this case, a producing processor is
10 always able to transmit a tuple to a requesting
processor by way of the link interfaces by which
they are connected.

15

20

25

30

35

WO 89/09967

PCT/US89/01456

21

Claims

1. A computer which comprises a plurality of processor modules, each having at least first and second I/O connection interfaces, a processor connected to those interfaces, and read-write memory, a common bus to which the first I/O connection interface of each processor module is connected, switch means connected to the second I/O connection interface of each processor module and operative to connect the second I/O connection interface of a selected processor module selectively to the second I/O connection interface of another processor module, and a switch operator connected to the bus for receiving over the bus data pertaining to a first processor module, which requires access to information, and to a second processor module, from which the required information is available, and for controlling the switch means to allow the required information to be transmitted from the first processor module to the second processor module by way of the switch means.

2. A computer according to claim 1, wherein the common bus is a parallel bus and the first I/O connection interface of each processor module is a parallel connection interface.

3. A computer according to claim 1, wherein the second I/O connection interface of each processor module is a link interface.

WO 89/09967

PCT/US89/01456

22

4. A computer according to claim 1, further comprising an external communication module which includes an external communication interface and is connected to the bus and to the switch means and is operative to transmit data between the external communication interface and the processor modules by way of the switch means or the bus.

5. A computer according to claim 4, wherein the external communication interface is a fiber optic transceiver.

6. A computer system which comprises at least first and second computer stations each comprising:

(a) a plurality of processor modules, each having at least first and second I/O connection interfaces, a processor connected to those interfaces, and read-write memory;

(b) a common bus to which the first I/O connection interface of each processor module is connected;

(c) switch means having a plurality connection interfaces connected to the second I/O connection interfaces of the plurality of processor modules respectively and also having at least one additional connection interface, the switch means being operative to connect any one of its connection interfaces to any of its other connection interfaces;

(d) a switch operator connected to the bus for receiving over the bus data pertaining to a first processor module, which requires access to information, and to a second processor module, from which the required information is available, and for controlling the switch means to allow the

WO 89/09967

PCT/US89/01456

23

required information to be transmitted from the first processor module to the second processor module by way of the switch means; and

5 (e) at least one external communication module which includes an external communication interface and is connected to the bus and to the switch means and is operative to transmit data between the external communication interface and the processor modules by way of the switch means or
10 the bus.

7. A computer system according to claim 6, wherein the external communication interface of an external communication module of the first computer
15 station is connected to the external communication interface of an external communication module of the second computer station.

8. A computer system according to claim 6, wherein the external communication interface is a
20 fiber optic transceiver.

9. A computer system according to claim 6, wherein the common bus of each computer is a
25 parallel bus and the first I/O connection interface of each processor module is a parallel connection interface.

10. A computer system according to claim 6, wherein the second I/O connection interface of each
30 processor module is a link interface and the additional connection interface of the switch means of each computer is a link interface.

35

WO 89/09967

PCT/US89/01456

24

11. A computer system according to claim 6, further comprising a user terminal having display means, data input means, at least one processor connected to the display means and data input means and having at least one I/O connection interface, and external communication means connected to the I/O connection interface and having an external communication interface which is connected to an external communication interface of at least one computer station for transmitting data between said one computer station and the processor of the user terminal.

12. A method of operating a computer comprising a plurality of processors each having at least first and second I/O connection interfaces and read-write memory, the method comprising:

(a) in the event that a first of said processors requires access to information that is not present in its memory, transmitting a request for information to at least a second of said processors by way of the first I/O connection interfaces of the first and second processors, and

(b) in the event that the required information is contained in the memory of the second processor, transmitting the required information from the second processor to the first processor by way of the second I/O connection interfaces of the first and second processors.

WO 89/09967

PCT/US89/01456

25

13. A method of operating a computer system which comprises at least first and second computer stations each comprising a plurality of processors each having at least first and second I/O connection interfaces and read-write memory, and each computer station also comprising external communication means, the method comprising:

(a) in the event that one processor of the first computer station requires access to information that is not present in the memory of any of the processors of the first computer station, transmitting a request for information to at least the second computer station by way of the first I/O connection interface of said one processor and the external communication means of the first and second computer stations, and

(b) in the event that the required information is contained in the memory of a processor of the second computer station, transmitting the required information from that processor to said one processor by way of the second I/O connection interface of the second processor, the external communication means of the second and first computer stations, and the second I/O connector interface of the first processor.

WO 89/89967

PCT/US89/01456

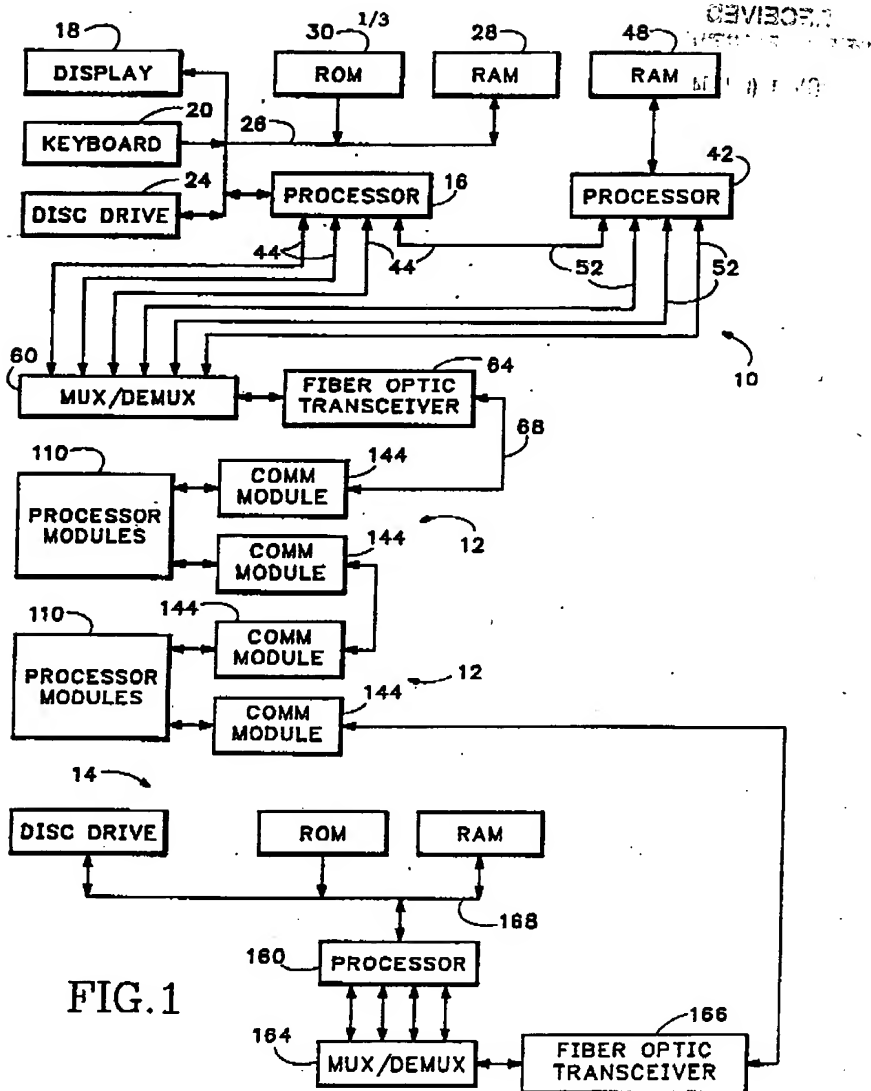


FIG. 1

WO 89/09967

PCT/US89/01456

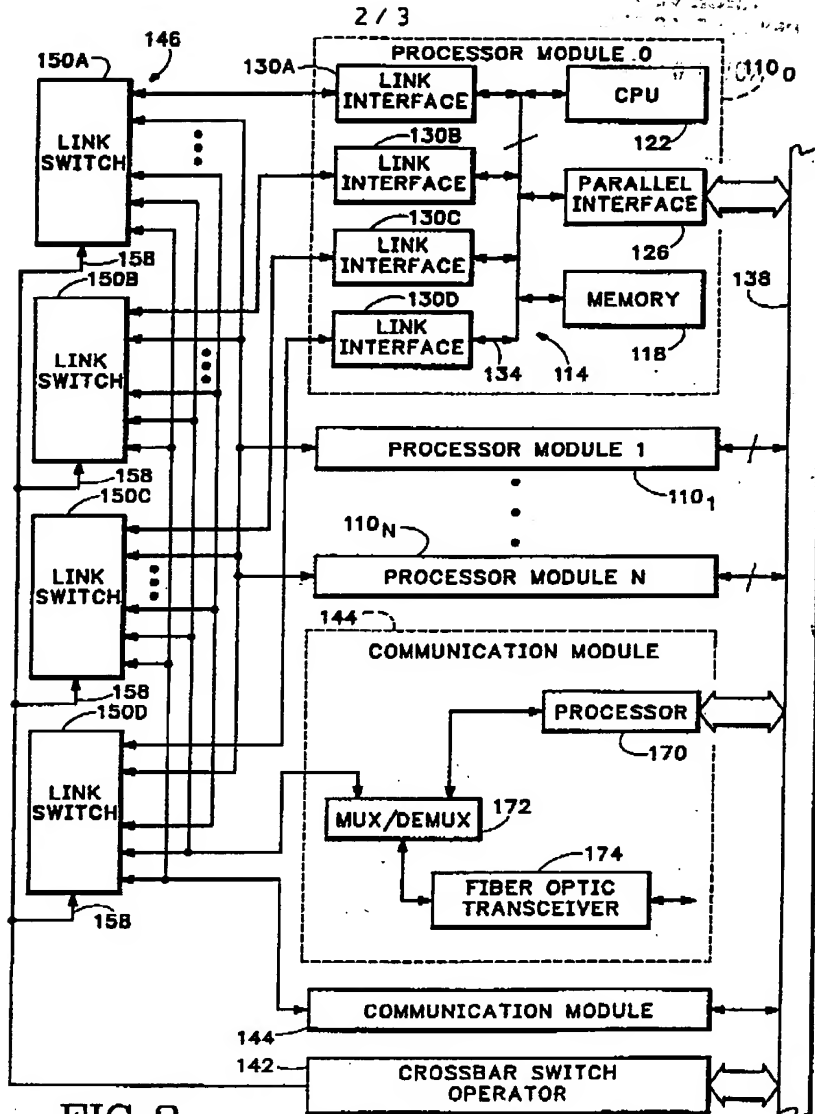


FIG. 2

SUBSTITUTE SHEET

WO 89/09967

PCT/US89/01456

3 / 3

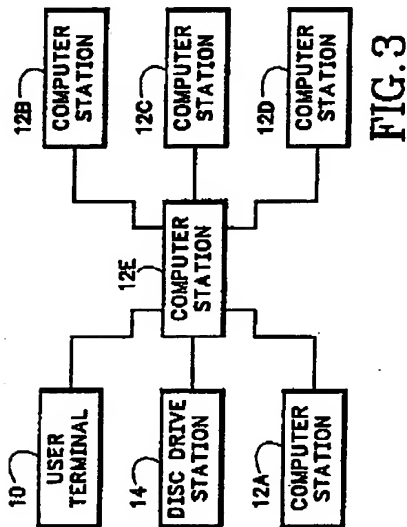


FIG. 3

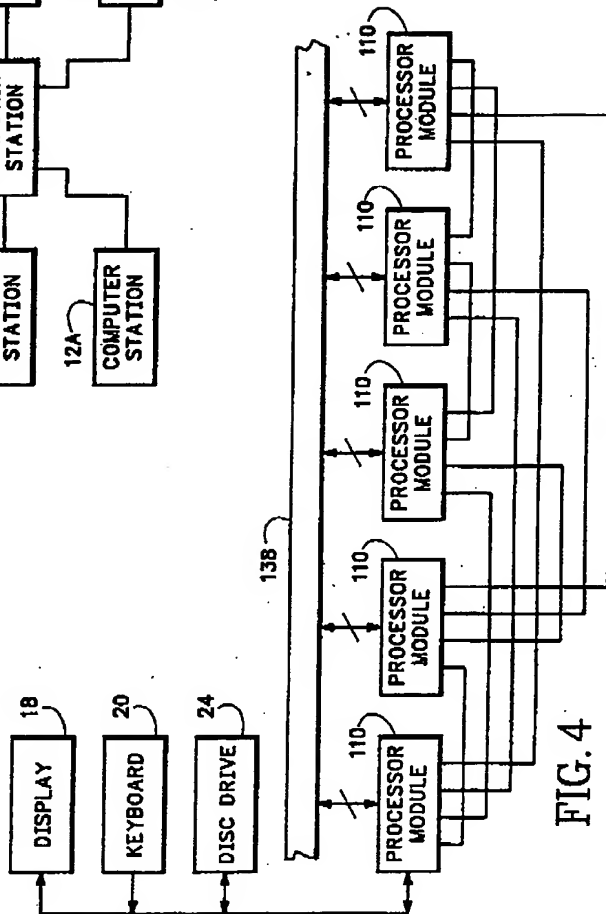


FIG. 4

SUBSTITUTE SHEET

INTERNATIONAL SEARCH REPORT

International Application No. PCT/US89/01456

I. CLASSIFICATION OF SUBJECT MATTER (If several classification symbols apply, indicate all ⁶)		
According to International Patent Classification (IPC) or to both National Classification and IPC		
IPC(4): G06F 13/38 G06F 13/42, G06F 15/56		
U.S. Cl. 364/200, 370/60		
II. FIELDS SEARCHED		
Minimum Documentation Searched ⁷		
Classification System	Classification Symbols	
U.S.	364/200 MS FILE, 364/900 MS FILE, 340/825.52 370/60, 370/67	
Documentation Searched other than Minimum Documentation to the extent that such Documents are included in the Fields Searched ⁸		
III. DOCUMENTS CONSIDERED TO BE RELEVANT ⁹		
Category ¹	Citation of Document, ¹¹ with indication, where appropriate, of the relevant passages ¹²	Relevant to Claim No. ¹³
Y	US, A,4,627,545 (Olson et al) 02 Dec., 1986 See figs. 1, 2, col. 7, lines 1 - 2, col. 7, lines 38 - 63, col. 8, lines 19 - 21	1 - 13
Y, P	US, A,4,811,210 (McAulay) 07 Mar., 1989 See figs. 3, 4, 5A, 16, col. 8, lines 14 - 34, col. 30, lines 16 - 22.	1 - 13
A	US, A,4,720,780 (Dolecek) 19 Jan., 1988	
A	US, A,4,644,496 (Andrews) 17 Feb., 1987	
A	US, A,4,417,334 (Gunderson et al) 22 Nov., 1983	
A, P	US, A,4,794,594 (Picard) 27 Dec., 1988	
A	US, A,4,633,473 (Ratchford et al) 30 Dec., 1986	
A, P	US, A,4,748,560 (Kataoka) 31 May, 1988	
<p>¹⁰ Special categories of cited documents: *</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier document but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> <p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.</p> <p>"Z" document member of the same patent family</p>		
IV. CERTIFICATION		
Date of the Actual Completion of the International Search	Date of Mailing of this International Search Report	
12 JUNE 1989	27 JUL 1989	
International Searching Authority	Signature of Authorized Officer	
ISA/US	THOMAS C. LEE <i>Thomas C. Lee</i>	

Form PCT/ISA/210 (second sheet) (Rev.11-87)